Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	54	(David near Glasco).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:07
L2	1022	711/141.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:07
L3	25486	"711"/$.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:07
L4	5604	cache near2 coheren$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:08
L5	8004	cache adj line	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:08
L6	6	remote adj cache adj controller	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:08
L7	114244	coherency or coherent or coherence	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:08
L8	214	remote adj directory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:08
L9	14415	MESI or MES or MEOSI	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:09

L10	2437	4 AND 5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:09
L11	6	6 AND 7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:09
L12	4	8 AND 9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:09
L13	1	10 AND 11	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:09
L14	0	12 AND 13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 14:10

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	51	first adj cache adj controller	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/05 15:03
S2	76	second$4 adj cache adj controller	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/05 15:03
S3	25	S1 and S2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/05 15:03
S4	1757	MESI	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/05 15:04
S5	1	S3 AND S4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/05 15:09
S6	2	"20040268052"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/05 15:09
S7	2	"20040268052".PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 09:50
S8	8661	cache adj line$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 09:51
S9	35	remote$2 adj2 cache adj2 controller$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 09:51

S10	117350	coheren$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 09:52
S11	61416	directory$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 09:52
S12	17	S8 and S9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 09:52
S13	4287	S10 and S11	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 09:52
S14	16	S12 and S13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 09:52
S15	1947	MESI or MOESI	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 09:52
S16	2	S14 AND S15	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 10:57
S17	2	"20030009623".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 11:26
S18	758039	remot$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 10:57
S19	2	S17 and S18	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 11:04

S20	680962	own$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 11:04
S21	0	S17 and S20	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 11:04
S22	2	"20030195939".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 11:27
S23	1	S20 and S22	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/07 11:27

Terms used
cache coherency read invalidate snoop cache line directory coherency coherent cache

Found **68** of **157,956**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 68 Result page: **1** 2 3 4 next

Relevance scale

1 Adaptive cache coherency for detecting migratory shared data

Alan L. Cox, Robert J. Fowler

May 1993 **ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture**, Volume 21 Issue 2

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references, citings, index terms

Parallel programs exhibit a small number of distinct data-sharing patterns. A common data-sharing pattern, migratory access, is characterized by exclusive read and write access by one processor at a time to a shared datum. We describe a family of adaptive cache coherency protocols that dynamically identify migratory shared data in order to reduce the cost of moving them. The protocols use a standard memory model and processor-cache interface. They do not require any compile-time or run-time ...

2 An evaluation of directory schemes for cache coherence

A. Agarwal, R. Simoni, J. Hennessy, M. Horowitz

May 1988 **ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture**, Volume 16 Issue 2

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

The problem of cache coherence in shared-memory multiprocessors has been addressed using two basic approaches: directory schemes and snoopy cache schemes. Directory schemes have been given less attention in the past several years, while snoopy cache methods have become extremely popular. Directory schemes for cache coherence are potentially attractive in large multiprocessor systems that are beyond the scaling limits of the snoopy cache schemes. Slight modifications to directory schemes can ...

3 An evaluation of directory schemes for cache coherence

Anant Agarwal, Richard Simoni, John Hennessy, Mark Horowitz

August 1998 **25 years of the international symposia on Computer architecture (selected papers)**

Full text available: pdf(1.31 MB) Additional Information: full citation, references, index terms

4 A performance evaluation of optimal hybrid cache coherency protocols

Jack E. Veenstra, Robert J. Fowler
September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems**, Volume 27 Issue 9
Full text available: pdf(1.28 MB) Additional Information: full citation, references, citings, index terms

5 Boosting the performance of hybrid snooping cache protocols
Fredrik Dahlgren
May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2
Full text available: pdf(1.23 MB) Additional Information: full citation, abstract, references, citings, index terms

Previous studies of bus-based shared-memory multiprocessors have shown hybrid write-invalidate/write-update snooping protocols to be incapable of providing consistent performance improvements over write-invalidate protocols. In this paper, we analyze the deficiencies of hybrid snooping protocols under release consistency, and show how these deficiencies can be dramatically reduced by using write caches and read snarfing.Our performance evaluation is based on program-driven simulation and a set o ...

6 Synchronization with multiprocessor caches
Joonwon Lee, Umakishore Ramachandran
May 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture**, Volume 18 Issue 3
Full text available: pdf(1.18 MB) Additional Information: full citation, abstract, references, citings, index terms

Introducing private caches in bus-based shared memory multiprocessors leads to the cache consistency problem since there may be multiple copies of shared data. However, the ability to snoop on the bus coupled with the fast broadcast capability allows the design of special hardware support for synchronization. We present a new lock-based cache scheme which incorporates synchronization into the cache coherency mechanism. With this scheme high-level synchronization primitives as well as low-le ...

7 Efficient synchronization primitives for large-scale cache-coherent multiprocessors
James R. Goodman, Mary K. Vernon, Philip J. Woest
April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the third international conference on Architectural support for programming languages and operating systems**, Volume 17 Issue 2
Full text available: pdf(1.48 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper proposes a set of efficient primitives for process synchronization in multiprocessors. The only assumptions made in developing the set of primitives are that hardware combining is not implemented in the inter-connect, and (in one case) that the interconnect supports broadcast. The primitives make use of synchronization bits (syncbits) to provide a simple mechanism for mutual exclusion. The proposed implementation of the primitives includes efficient (i.e.

8 The sun fireplane system interconnect
Alan Charlesworth
November 2001 **Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: pdf(224.87 KB) Additional Information: full citation, abstract, references, citings, index terms

System interconnect is a key determiner of the cost, performance, and reliability of large

cache-coherent, shared-memory multiprocessors. Interconnect implementations have to accommodate ever greater numbers of ever faster processors. This paper describes the Sun™ Fireplane two-level cache-coherency protocol, and its use in the medium and large-sized UltraSPARC-III-based Sun Fire™ servers.

9 C²MP: a cache-coherent, distributed memory multiprocessor-system

D. E. Marquardt, H. S. Alkhatib

August 1989 **Proceedings of the 1989 ACM/IEEE conference on Supercomputing**

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references, citings, index terms

Current research into the problems of cache coherency in multiprocessor (MP) systems, has primarily focused on bus based memory interconnection networks (M-ICN) and the use of various types of "snooping" cache coherency protocols. Bus bandwidth limitations can be alleviated through the use of wider bandwidth general interconnection structures, such as a crossbar switch. However, if private caches are used, the cache coherency problem becomes mul ...

10 Cache coherence in large-scale shared-memory multiprocessors: issues and comparisons

David J. Lilja

September 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 3

Full text available: pdf(3.12 MB) Additional Information: full citation, references, citings, index terms

11 Measuring memory hierarchy performance of cache-coherent multiprocessors using micro benchmarks

Cristina Hristea, Daniel Lenoski, John Keen

November 1997 **Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available: pdf(97.47 KB) Additional Information: full citation, abstract, references, citings

Even with today's large caches, the increasing performance gap between processors and memory systems imposes a memory bottleneck for many important scientific and commercial applications. This bottleneck is intensified in shared-memory multiprocessors by contention and the effects of cache coherency. Under heavy memory contention, the memory latency may increase 2 or 3 times. Nonethless, as more sophisticated techniques are used to hide latency and increase bandwidth, measuring memory performanc ...

12 A distributed shared memory multiprocessor ASURA: memory and cache architecture

S. Mori, H. Saito, M. Goshima, S. Tomita, M. Yanagihara, T. Tanaka, D. Fraser, K. Joe, H. Nitta

December 1993 **Proceedings of the 1993 ACM/IEEE conference on Supercomputing**

Full text available: pdf(1.17 MB) Additional Information: full citation, references, citings, index terms

13 Evaluation of the lock mechanism in a snooping cache

Toshiaki Tarui, Takayuki Nakagawa, Noriyasu Ido, Machiko Asaie, Mamoru Sugie

August 1992 **Proceedings of the 6th international conference on Supercomputing**

Full text available: pdf(1.11 MB) Additional Information: full citation, abstract, references, index terms

This paper discusses the design concepts of a lock mechanism for a Parallel Inference Machine (the PIM/c prototype) and investigates the performance of the mechanism in detail. Lock operations are extremely frequent on the PIM; however, lock contention rarely occurs during normal memory usage. For this reason, the lock mechanism is designed so as

to minimize the lock overhead time in the case of no contention. This is done by using an invalidation lock mechanism, which utilizes t ...

14 Verification of the Futurebus+ cache coherence protocol: a case study in model checking

Kylie Williams, Robert Esser

January 2004 **Proceedings of the 27th conference on Australasian computer science - Volume 26**

Full text available: pdf(175.99 KB) Additional Information: full citation, abstract, references

This paper presents a case study for automatic verification using the Communicating Sequential Processes formalism. The case study concerns the Futurebus+ cache coherency standard; we develop a formal model of the protocol and perform some verification tasks upon it. In the process of doing so, we extend the previous solution by developing a formal specification of cache coherence that is suitable for the verification of both directory and snooping based cache coherence protocols.

15 Symmetric Multiprocessing on Programmable Chips Made Easy

Austin Hung, William Bishop, Andrew Kennings

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 1**

Full text available: pdf(181.58 KB) Additional Information: full citation, abstract

Vendor-provided softcore processors often support advanced features such as caching that work well in uniprocessor or uncoupled multiprocessor architectures. However, it is achallenge to implement Symmetric Multiprocessor on a Programmable Chip (SMPoPC) systems using such processors. This paper presents an implementation of a tightly-coupled, cache-coherent symmetric multiprocessing architecture using a vendor-provided softcore processor. Experimental results show that this implementation can be ...

16 Comparison of hardware and software cache coherence schemes

Sarita V. Adve, Vikram S. Adve, Mark D. Hill, Mary K. Vernon

April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture**, Volume 19 Issue 3

Full text available: pdf(1.22 MB) Additional Information: full citation, references, citings, index terms

17 Design and performance of a coherent cache for parallel logic programming architectures

A. Goto, A. Matsumoto, E. Tick

April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture**, Volume 17 Issue 3

Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes the design and performance of a tightly-coupled shared-memory coherent cache optimized for the execution of parallel logic programming architectures. The cache utilizes a copy-back write-allocation protocol having five states and a hardware lock mechanism. Optimizations for logic programming are introduced in four software-controlled memory access commands: direct-write, exclusive-read, read-purge, and read-invalidate. In this paper we describe these operations and pres ...

18 Verification techniques for cache coherence protocols

Fong Pong, Michel Dubois

March 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 1

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references, citings, index terms

In this article we present a comprehensive survey of various approaches for the verification of cache coherence protocols based on state enumeration, (symbolic model checking, and symbolic state models. Since these techniques search the state space of the protocol exhaustively, the amount of memory required to manipulate that state information and the verification time grow very fast with the number of processors and the complexity of the protocol mechanism ...

Keywords: cache coherence, finite state machine, protocol verification, shared-memory multiprocessors, state representation and expansion

19 STiNG: a CC-NUMA computer system for the commercial marketplace

Tom Lovett, Russell Clapp

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2

Full text available: pdf(1.30 MB) Additional Information: full citation, abstract, references, citings, index terms

"STiNG" is a Cache Coherent Non-Uniform Memory Access (CC-NUMA) Multiprocessor designed and built by Sequent Computer Systems, Inc. It combines four processor Symmetric Multi-processor (SMP) nodes (called Quads), using a Scalable Coherent Interface (SCI) based coherent interconnect. The Quads are based on the Intel P6 processor and the external bus it defines. In addition to 4 P6 processors, each Quad may contain up to 4 GBytes of system memory, 2 Peripheral Component Interface (PCI) busses for ...

20 A memory management unit and cache controller for the MARS system

Feipei Lai, Chyuan-Yow Wu, Tai-Ming Parng

November 1990 **Proceedings of the 23rd annual workshop and symposium on Microprogramming and microarchitecture**

Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, references

For large caches, the interaction between cache access and address translation affects the machine cycle time and the access time to memory. The physically addressed caches slow down the cache access due to the virtual address translation. The virtually addressed caches is faster, but the synonym problem is difficult to handle. By some software constraints and hardware support, our virtually addressed physically tagged caches can achieve the same speed as traditional virtually addressed cac ...